

# Soutenance CDA

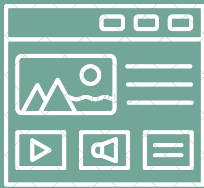
Chaouche Malek

2024

## 1. Structure du Titre CDA

Le titre est composé de 3 certificats de compétences professionnelles :

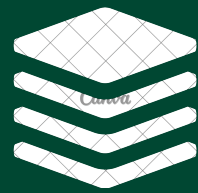
CONCEVOIR ET  
DÉVELOPPER DES  
COMPOSANTS  
D'INTERFACE  
UTILISATEUR EN  
INTÉGRANT LES  
RECOMMANDATIONS DE  
SÉCURITÉ



CONCEVOIR ET  
DÉVELOPPER LA  
PERSISTENCE DES  
DONNÉES EN  
INTÉGRANT LES  
RECOMMANDATIONS DE  
SÉCURITÉ

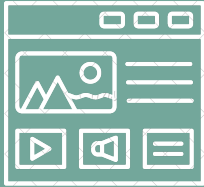


CONCEVOIR ET  
DÉVELOPPER UNE  
APPLICATION  
MULTICOUCHE  
RÉPARTIE EN  
INTÉGRANT LES  
RECOMMANDATIONS DE  
SÉCURITÉ



⊖ Il est primordial d'avoir un projet qui couvre les compétences suivantes : ⊖

CONCEVOIR ET  
DÉVELOPPER DES  
COMPOSANTS  
D'INTERFACE  
UTILISATEUR EN  
INTÉGRANT LES  
RECOMMANDATIONS DE  
SÉCURITÉ



MAQUETTAGE DE L'APPLICATION

DÉVELOPPER DES COMPOSANTS D'ACCÈS AUX  
DONNÉES

DÉVELOPPER LA PARTIE FRONT-END D'UNE  
INTERFACE UTILISATEUR WEB

DÉVELOPPER LA PARTIE BACK-END D'UNE  
INTERFACE UTILISATEUR WEB

CONCEVOIR ET  
DÉVELOPPER LA  
PERSISTANCE DES  
DONNÉES EN INTÉGRANT  
LES RECOMMANDATIONS  
DE SÉCURITÉ



CONCEPTION DE LA BASE DE DONNÉES

MISE EN PLACE SUITE À LA CONCEPTION

CONCEVOIR ET  
DÉVELOPPER UNE  
APPLICATION  
MULTICOUCHE RÉPARTIE  
EN INTÉGRANT LES  
RECOMMANDATIONS DE  
SÉCURITÉ



CONCEVOIR UNE APPLICATION

DÉVELOPPER DES COMPOSANTS MÉTIER

CONSTRUIRE UNE APPLICATION ORGANISÉE EN  
COUCHES

## 2. Attentes du jury

**Le jury attend du concepteur développeur d'application qu'il possède de solides compétences en programmation, qu'il soit capable de concevoir et de développer des applications fonctionnelles répondant aux besoins des utilisateurs. Il doit également être capable de travailler en équipe, de communiquer efficacement et de résoudre les problèmes de manière autonome. La connaissance des dernières technologies et tendances du marché est également un atout important.**

**Enfin, le jury attend du candidat qu'il soit passionné, motivé et prêt à apprendre en continu.**

### Maquettage de l'application

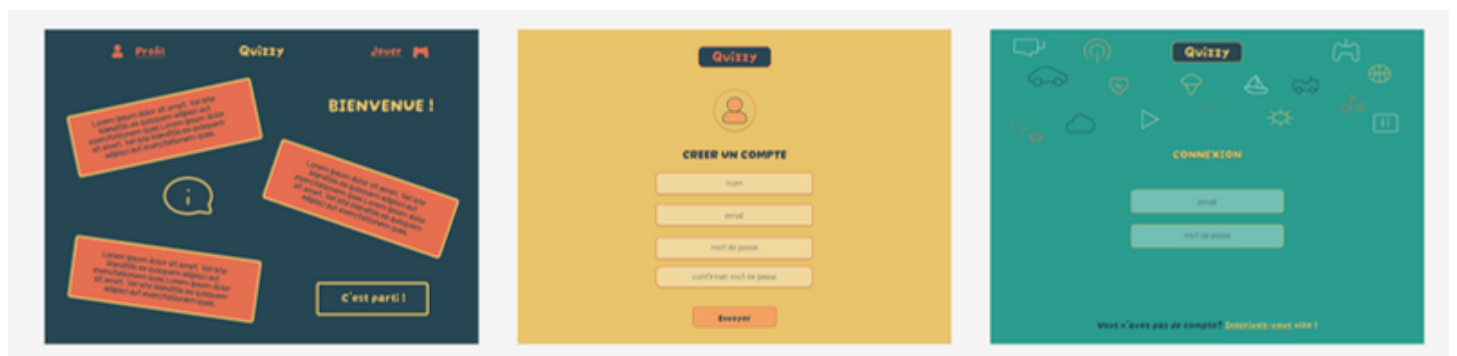
Votre application doit satisfaire les besoins de l'utilisateur et suivre les bonnes pratiques UX/UI. Vous devriez être en mesure de fournir une maquette de l'application demandée à vos clients.

Pour l'examen, vous devrez créer une maquette responsive pour plusieurs écrans, incluant les versions mobiles et desktop. Il est recommandé de concevoir l'application "mobile first" afin de la rendre adaptable aux différentes tailles d'écran.

**Quelques exemples :**

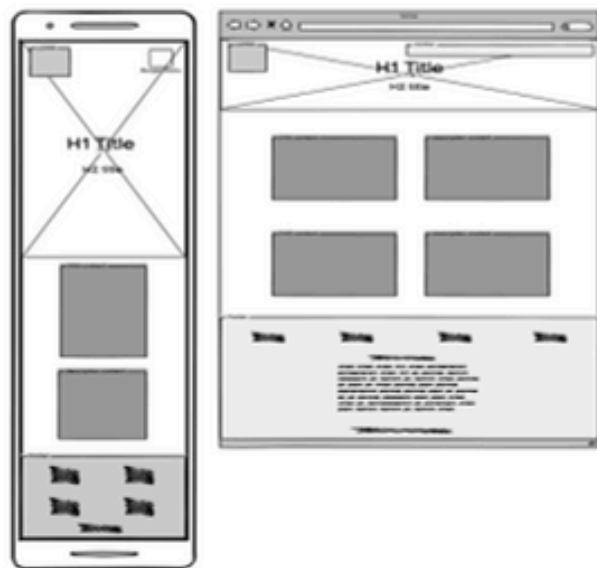


Version mobile

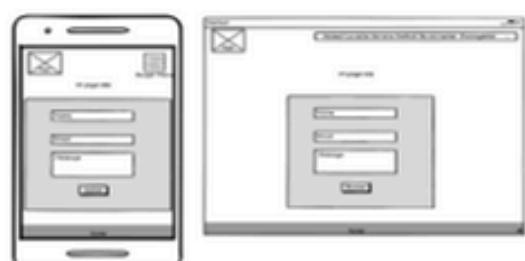
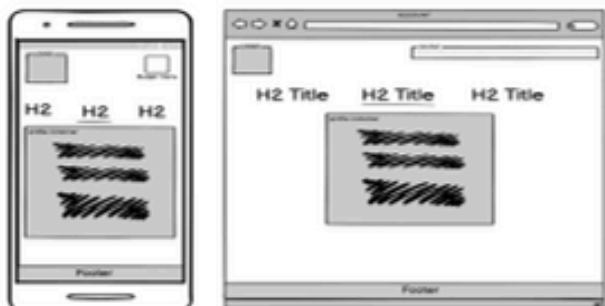
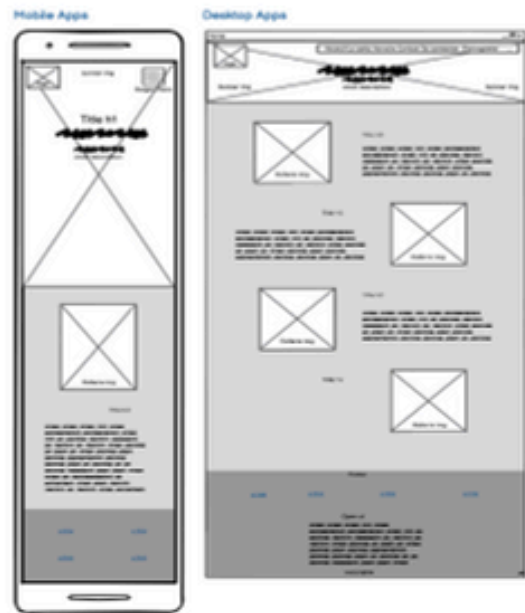


Version desktop

## ZONING



## WIREFRAMES



Les maquettes ne sont pas toujours graphiquement déterminantes. Comme dans l'exemple ci-dessus un simple wireframe générique sans design graphique peut suffire.

# Charte graphique

Toujours dans un soucis de satisfaction utilisateur, vous devez proposer une identité graphique pour votre application.

Rien de mieux que de produire une charte graphique qui représentera l'identité visuelle de votre application.

Si je vous dis coca cola, vous imaginez tout de suite le rouge blanc comme couleur et le style de police *Spencerian Script*

Votre application web, aura une identité visuelle, tant par ses couleurs dominantes que les typographies, les icônes et les éléments visuels. Elle donne une cohérence esthétique à l'application et reflète souvent l'image de marque de l'entreprise ou du projet.

Elaborer une charte graphique fait partie de la phase de conception et doit faire partie intégrante de votre projet.

**Quelques exemples :**

# Chartre Graphique

## Font-family:

Titre h1: **Darumadrop One**

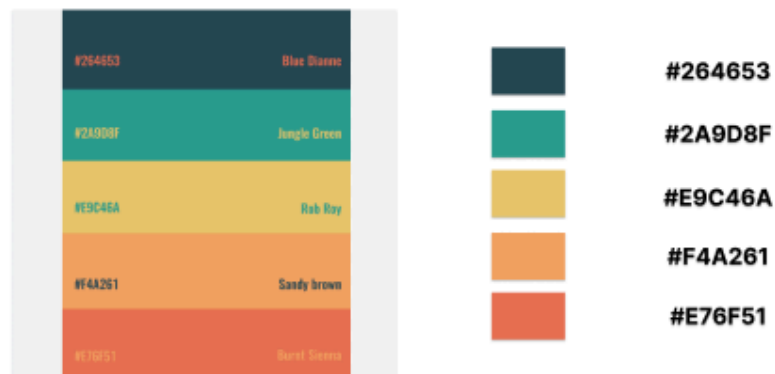
Paragraphe: Pangolin

Logo: **Darumadrop One**

## Logo:



## Palette de couleurs:





**Font-Family : Title :**

'Edwardian Scr Alt ITC TT', 'Luxurious Script', 'Balletregular', sans-serif

**Paragraphe :**

(Bootstrap):system-ui,-apple-system,"Segoe UI",Roboto,"Helvetica Neue","Noto Sans","Liberation Sans",Arial,sans-serif,"Apple Color Emoji","Segoe UI Emoji","Segoe UI Symbol","Noto Color Emoji";

Font-Family-Logo : Edwardian Script ITC : *Le Quai Antique*

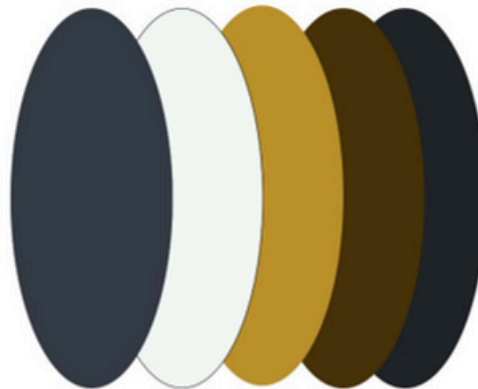
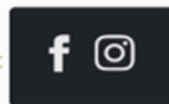
**Logo :**



**Couleurs :**



**Sociaux :**



## Conception de l'application

Les membres du jury auront de plus grandes attentes sur la partie conception de votre application, plutôt que sur la partie fonctionnelle en elle même.

L'idée c'est de comprendre comment vous avez fait pour partir d'une feuille blanche, d'un besoin clients, pour ensuite aboutir en la création d'une application web.



Parmi les différentes phases de conception, une des premières étapes concerne l'élaboration d'un diagramme de cas d'utilisation (use cases)

## Diagramme de cas d'utilisation

*un élément clé de la conception d'applications web*

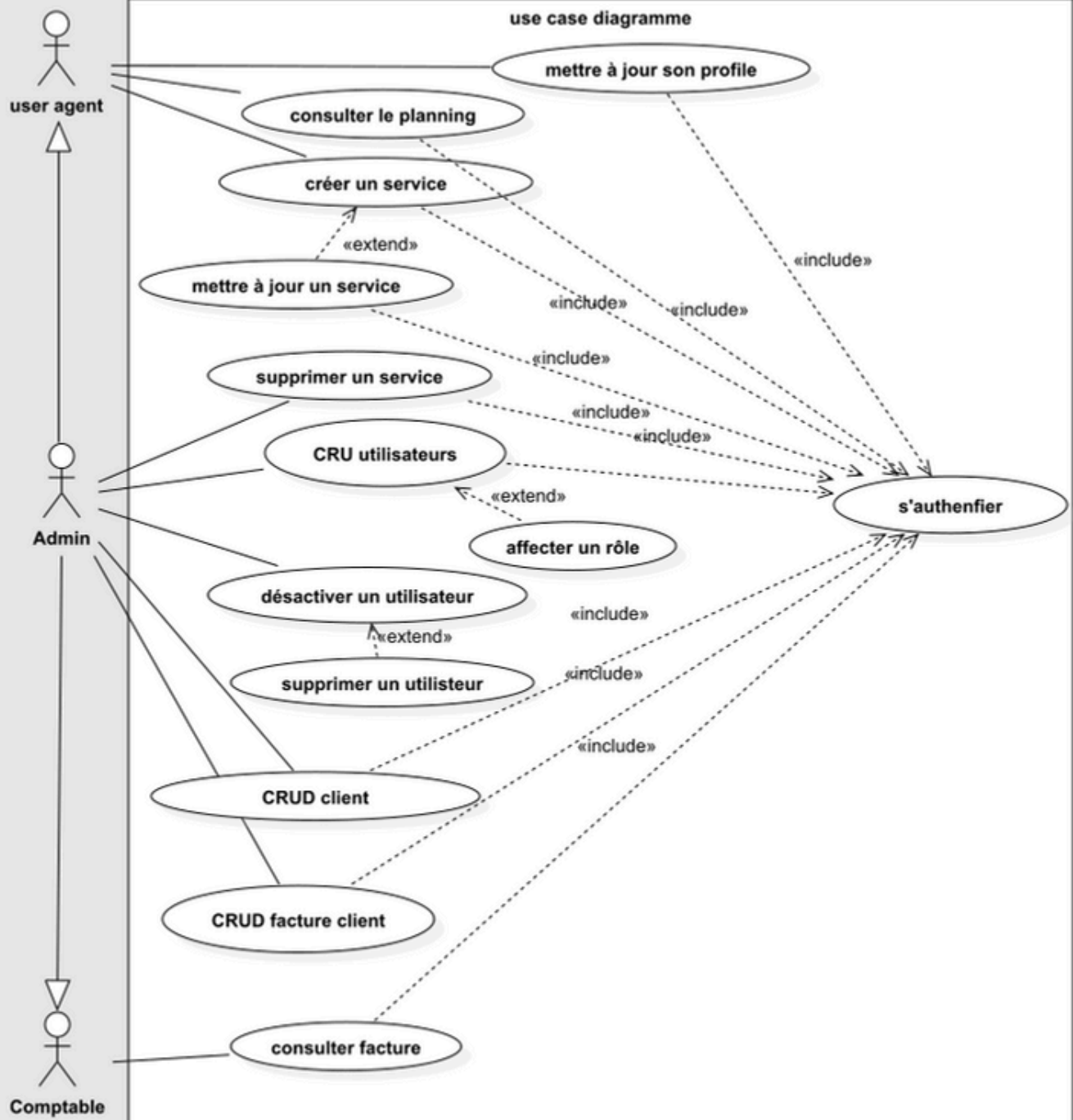
Le diagramme de cas d'utilisation est une technique de modélisation qui décrit les interactions entre les utilisateurs (acteurs) et le système (l'application web). Il permet de comprendre les fonctionnalités requises et les objectifs de l'application. Les concepteurs peuvent hiérarchiser les fonctionnalités, identifier les besoins des utilisateurs, définir les scénarios d'utilisation et spécifier les exigences du système en utilisant le diagramme de cas d'utilisation. Cela donne une vision claire des fonctionnalités à intégrer et aligne la conception de l'application sur les besoins des utilisateurs.

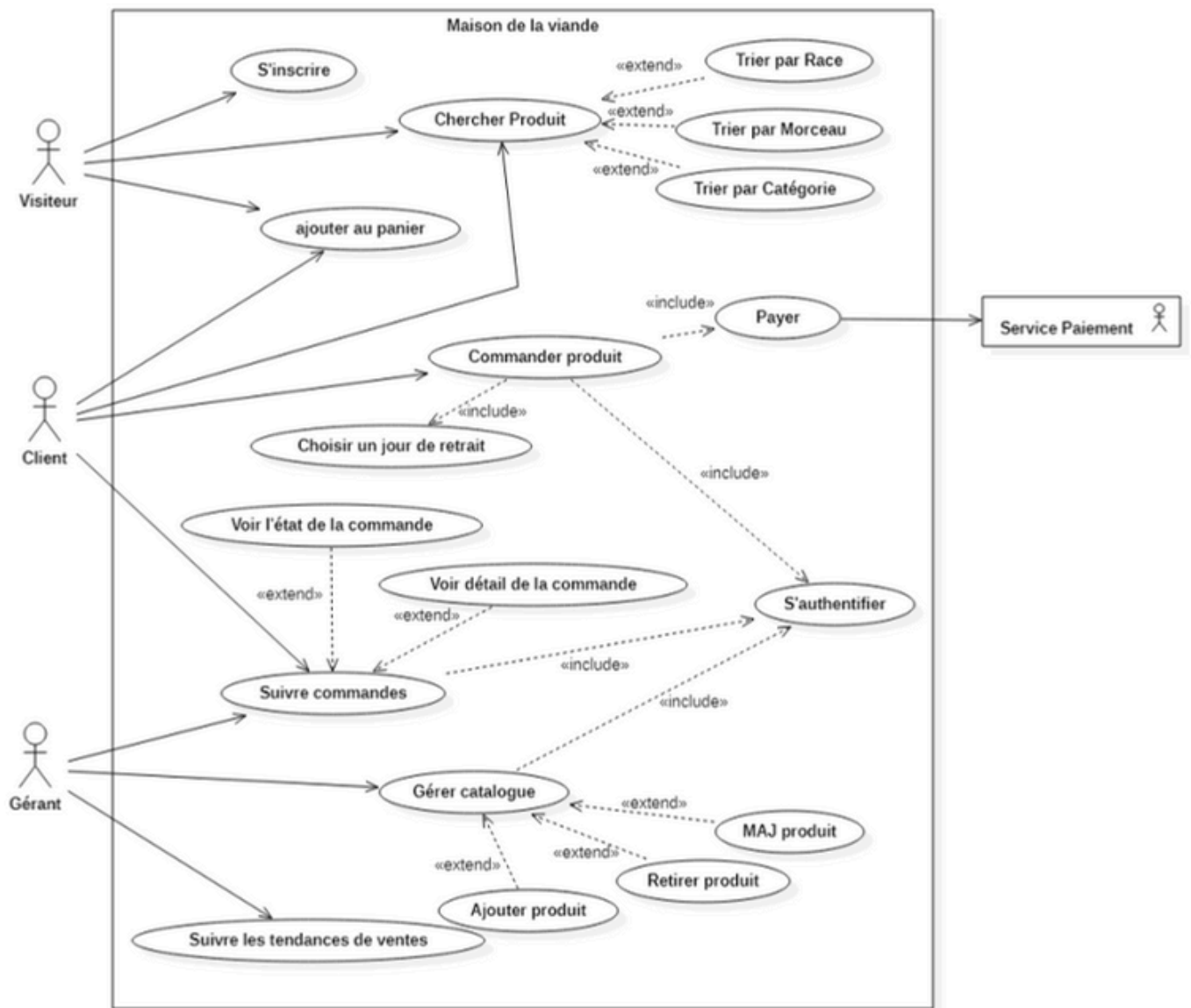
Le diagramme de cas d'utilisation sert de base pour d'autres activités de conception, telles que la création de maquettes d'interface utilisateur, la spécification des interactions et la définition des exigences fonctionnelles pour le développement de l'application web. En somme, il est un élément clé de la conception d'applications web.

**Quelques exemples :**

côté admin(le gérant)







## Diagramme de séquence

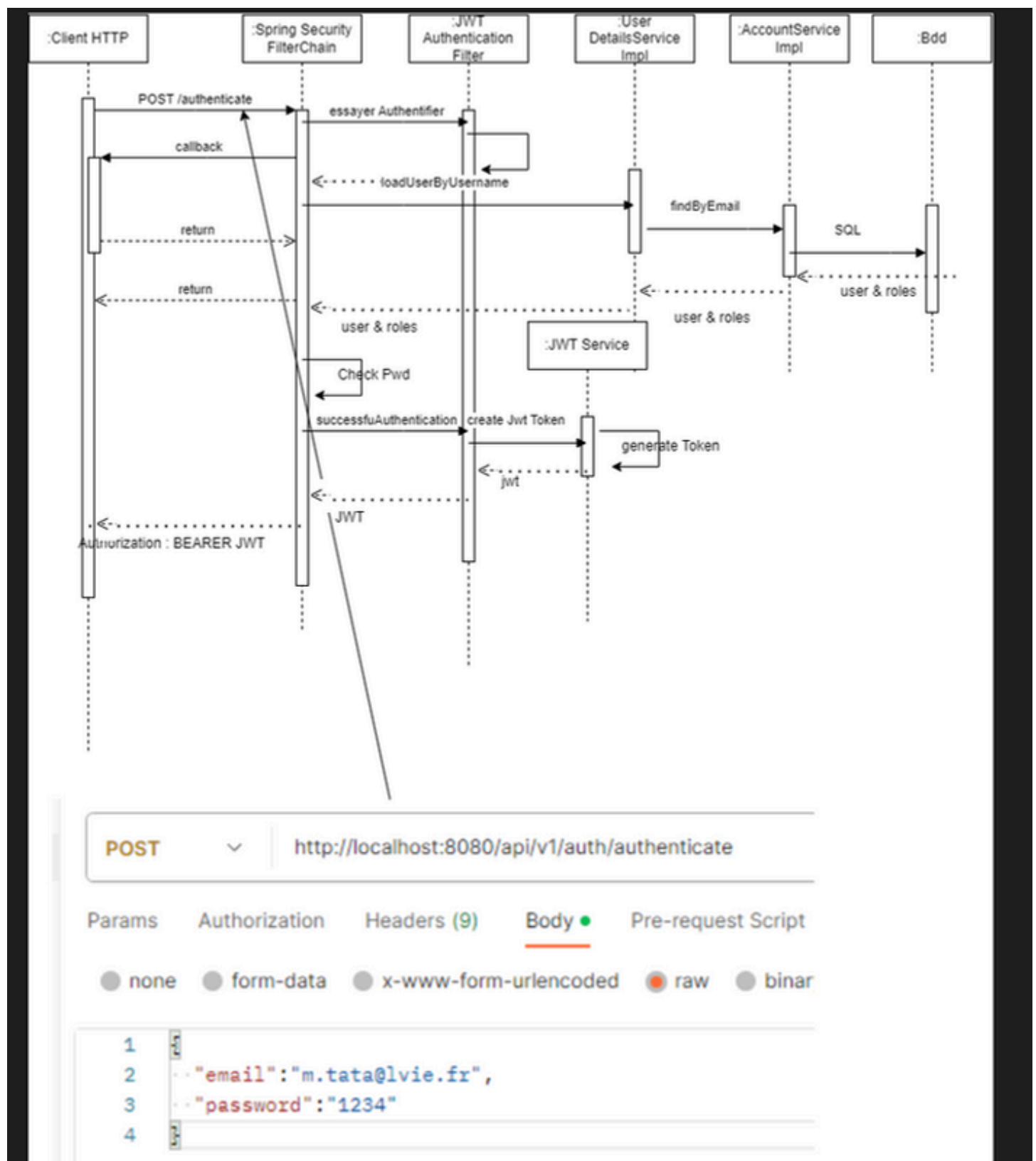
Le diagramme de séquence fait également partie de la phase de conception dans la création d'une application web. Il est utilisé pour modéliser les interactions entre les objets ou les composants du système pendant l'exécution d'un scénario spécifique.

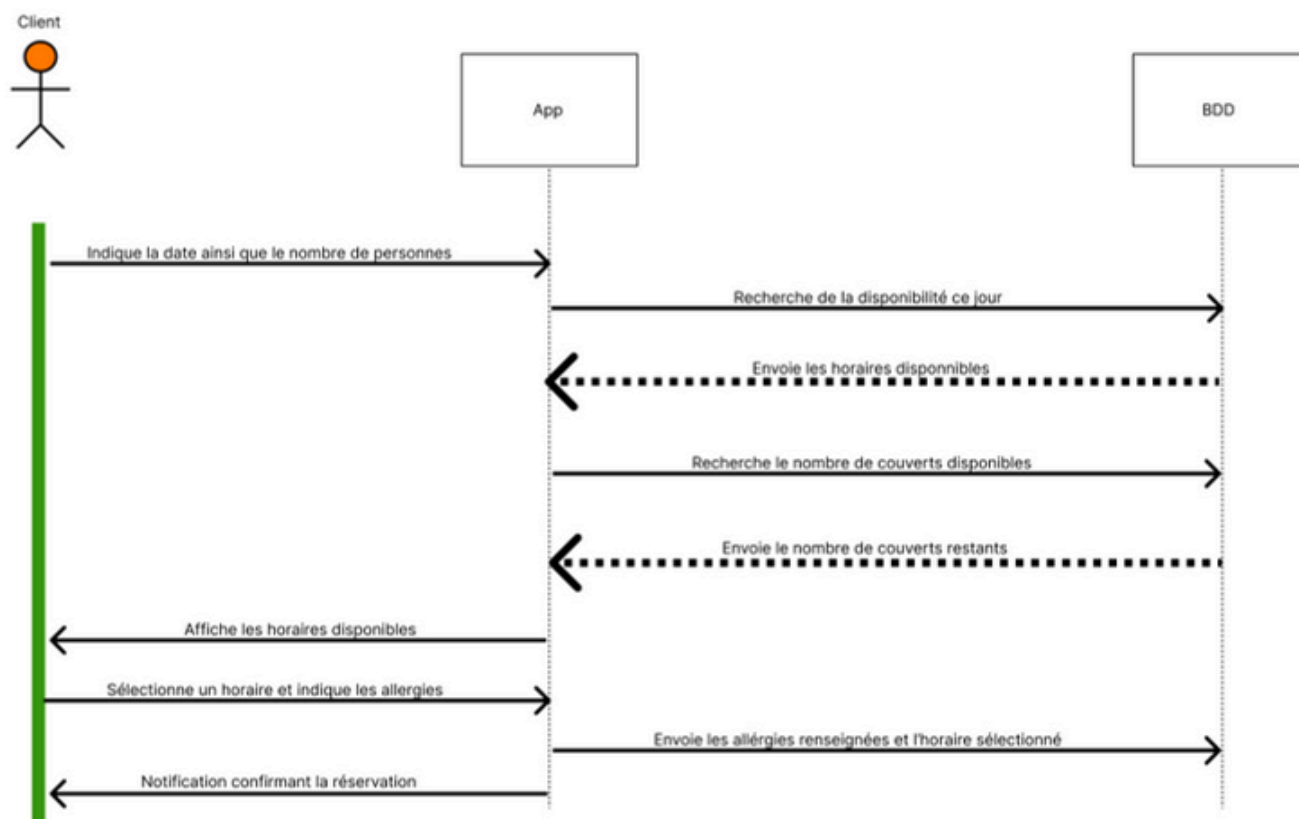
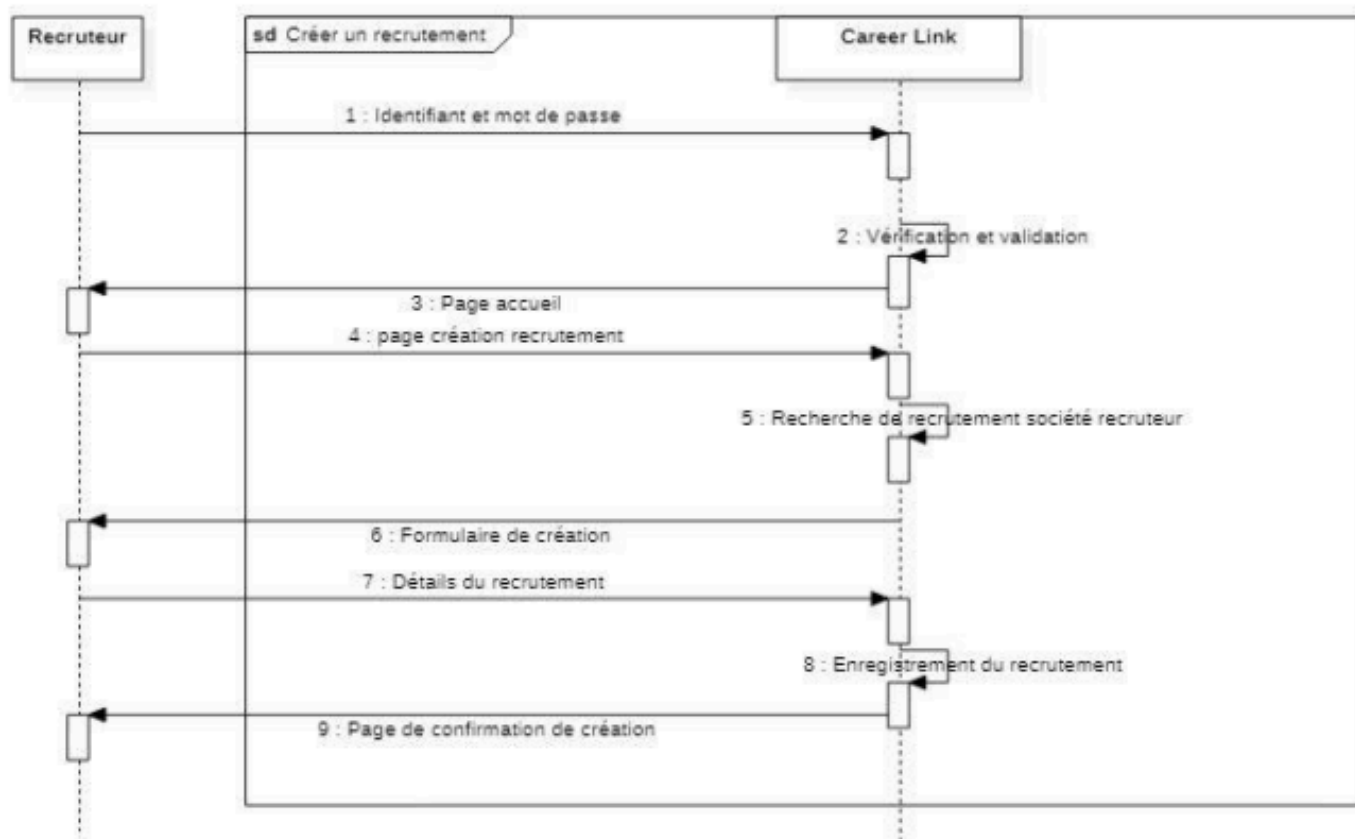
Dans la phase de conception, une fois que les besoins ont été analysés et que les cas d'utilisation ont été identifiés, le diagramme de séquence peut être créé pour détailler l'ordre chronologique des messages échangés entre les objets ou les composants du système lors de l'exécution d'un cas d'utilisation.

Le diagramme de séquence montre comment les objets interagissent entre eux au fil du temps pour accomplir une fonctionnalité spécifique. Il met en évidence

les messages échangés, les appels de méthodes, les retours et les synchronisations entre les différents acteurs et objets.

### Quelques exemples :





## Schéma de la base de données

Avant de pouvoir élaborer une base de données dans le code, il est nécessaire de penser à sa structure au préalable.

Pour l'examen, il va être attendu la fourniture d'une conception de la base de données. Vous avez 2 possibilités :

- la méthodologie Merise
- le diagramme de classe

Personnellement, je recommande vivement la méthodologie Merise.

Cette méthodologie de conception et de modélisation des systèmes d'information a été développée en France dans les années 1970 et est largement utilisée pour concevoir des bases de données et des systèmes d'information.

En effet, elle propose une approche structurée et méthodique pour concevoir et développer des systèmes d'information. Elle se base sur une division en trois niveaux : le niveau conceptuel, le niveau organisationnel et le niveau physique.

Le niveau conceptuel (Modèle Conceptuel des Données - MCD) permet de modéliser les concepts métier et les relations entre les données de manière abstraite.

Le niveau physique (Modèle Physique des Données - MPD) permet de concevoir la structure détaillée de la base de données et de spécifier les contraintes techniques.

**Quelques exemples :**





C'est de loin la partie la plus complexe car souvent méconnue par les candidats. "En quoi votre application est multicouche" ? c'est la question pertinente qu'il faut se poser car sa réponse peut vous donner un bon indicateur quand à la conformité de votre projet vis à vis des attentes du CCP en question.

Une architecture multicouche est généralement utilisée pour séparer les différentes responsabilités et améliorer la maintenabilité et la scalabilité de l'application. Une application multicouche est composée de différentes couches qui interagissent entre elles de manière organisée.

### **Exemple avec une API RESTFUL JAVA avec SPRING**

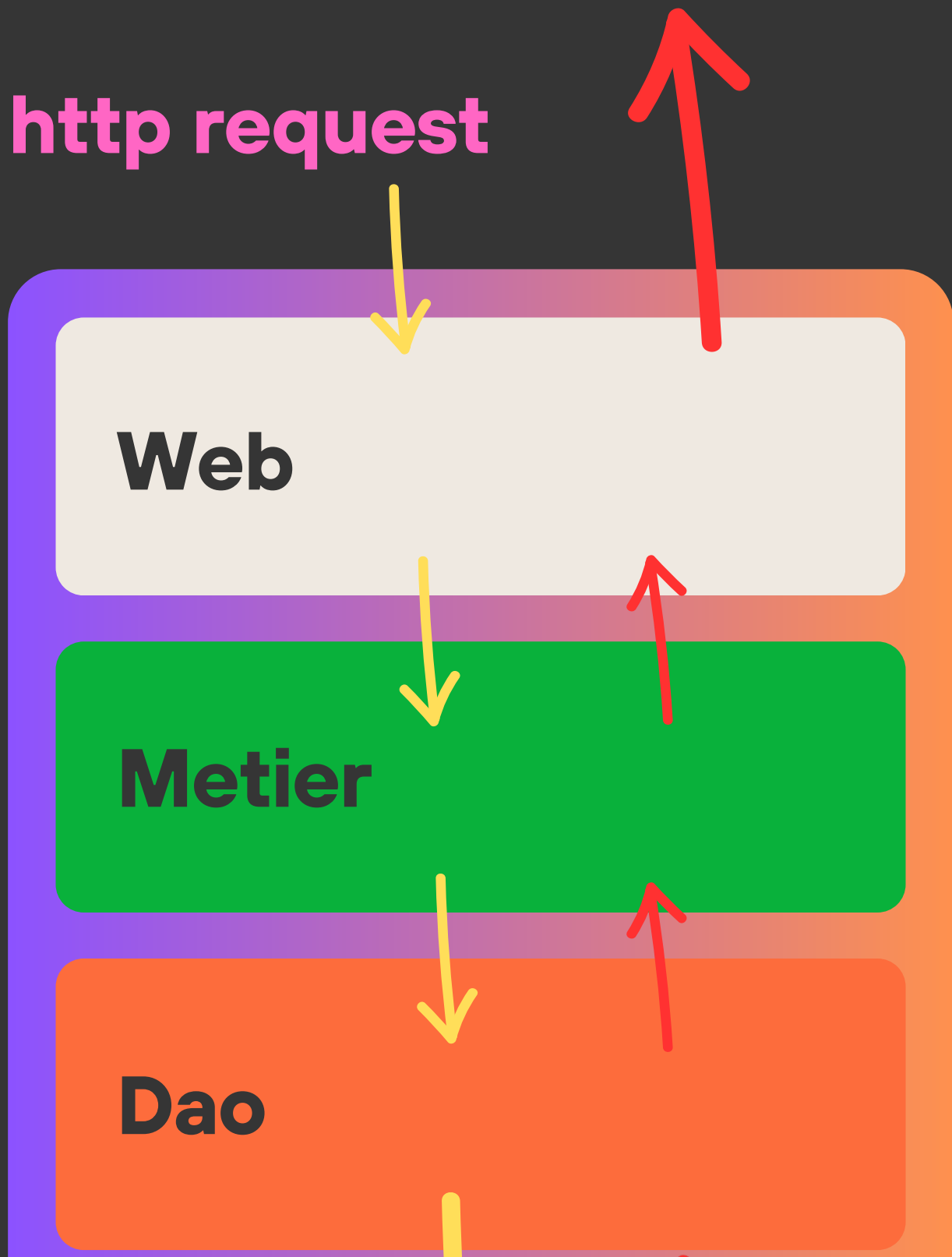
Une bonne idée d'application côté back end est la création d'une api RESTFUL. Java est d'ailleurs essentiellement utilisé aujourd'hui dans des projets sous la forme de micro services Rest.

Dans cette configuration nous aurions une architecture réparties en plusieurs couches comme ceci :



# http response

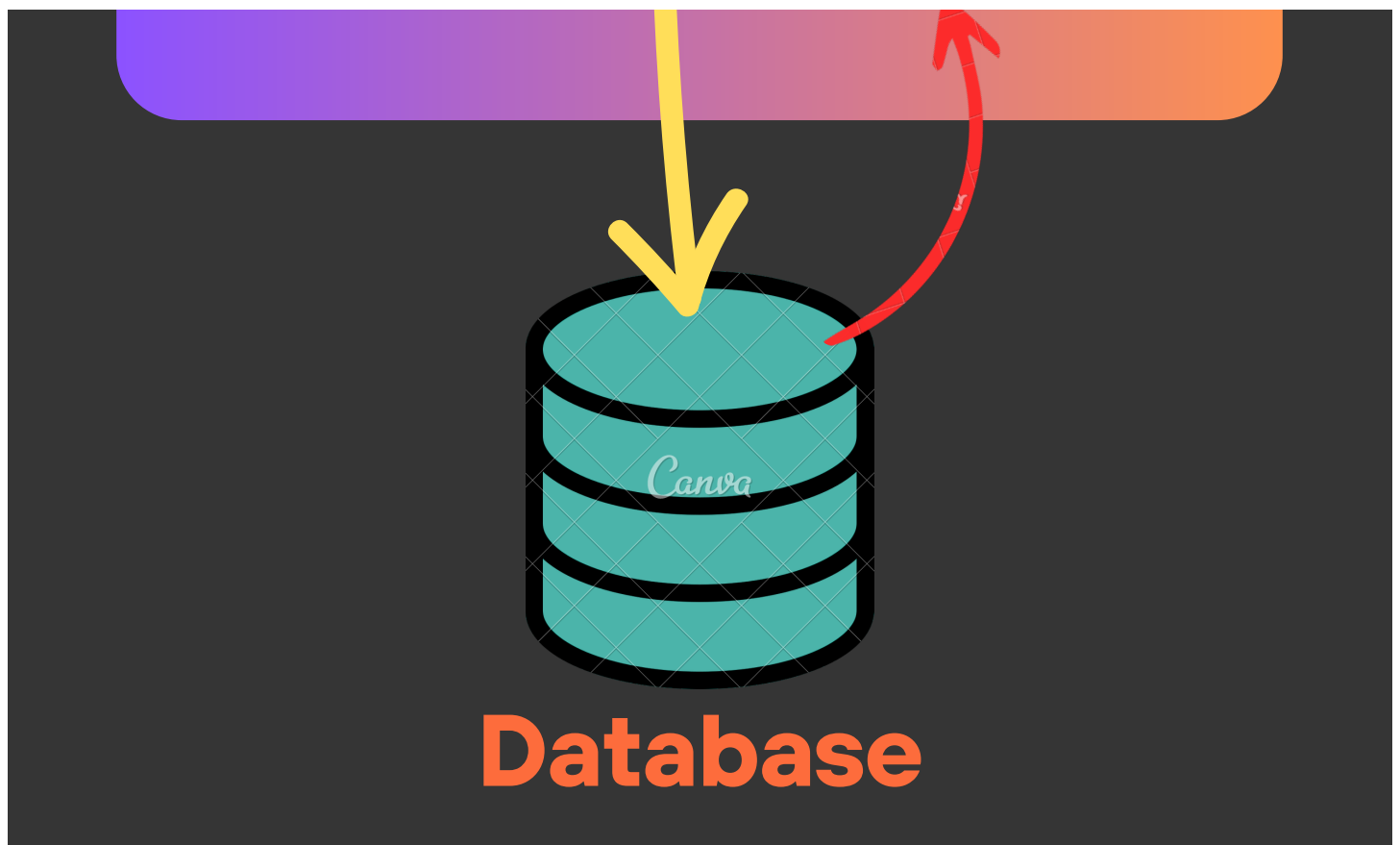
# http request



# Web

# Metier

# Dao



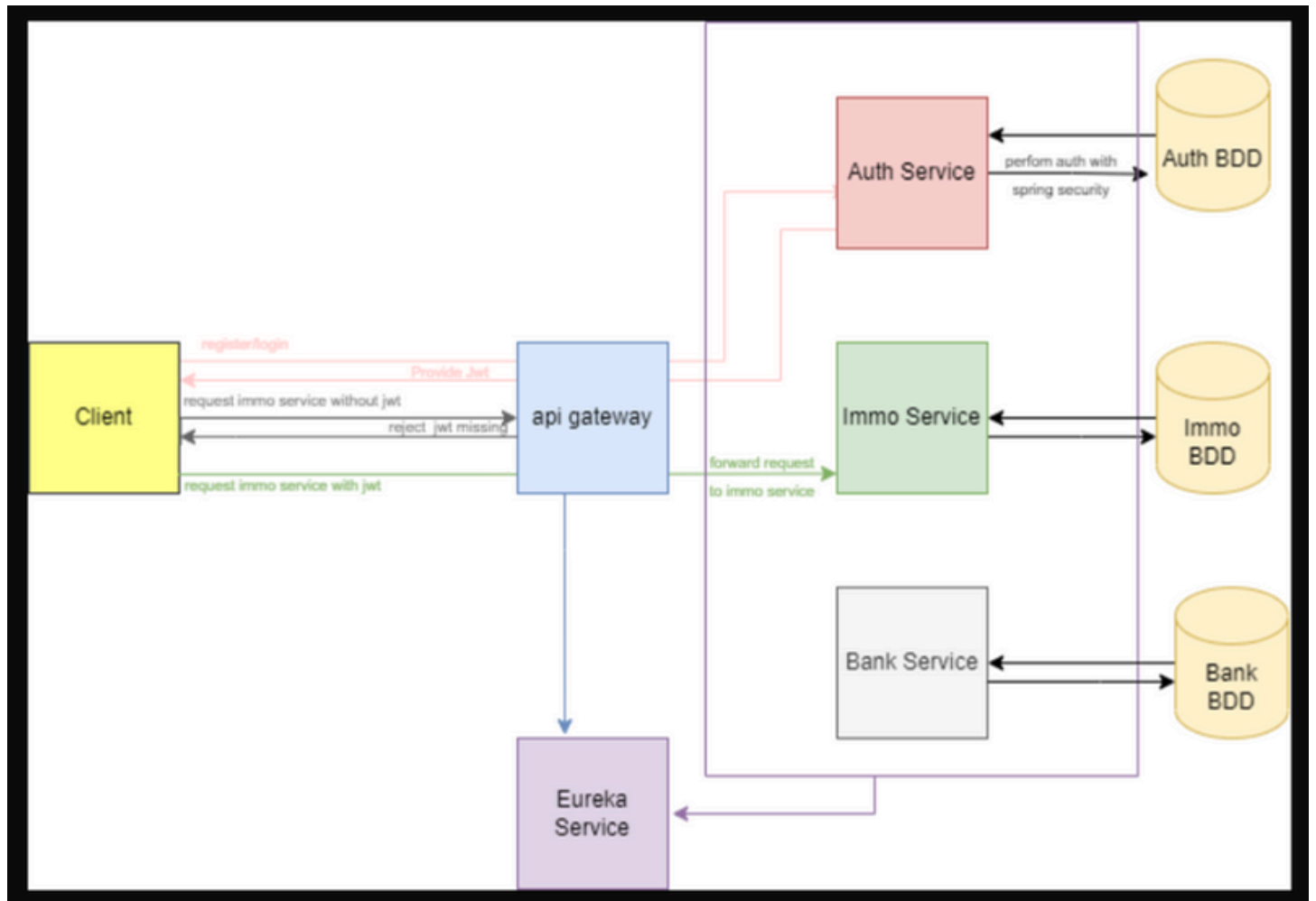
**Couche de présentation :** Cette couche comprend les RestControllers qui exposent les points d'entrée de votre API RESTful. Ces composants reçoivent les requêtes HTTP, traitent les paramètres, effectuent la validation et renvoient les réponses appropriées.

**Couche métier (service) :** Les services (ou business ) représentent la logique métier de votre application. Ils encapsulent les opérations complexes et l'interaction entre plusieurs entités. Les services peuvent effectuer des opérations telles que la validation des données, la transformation des objets, l'appel à d'autres services ou composants, etc.

**Couche d'accès aux données :** Les repositories (ou DAO - Data Access Objects) font partie de cette couche. Ils sont responsables de l'accès aux données persistantes, généralement via une base de données. Les repositories fournissent des méthodes pour effectuer des opérations de création, de lecture, de mise à jour et de suppression (CRUD) sur les entités de la base de données.

**Couche de persistance des données** : Cette couche concerne la base de données elle-même, où les données sont stockées de manière persistante. Elle peut être constituée d'un système de gestion de base de données (SGBD) tel que MySQL, PostgreSQL, MongoDB, etc.

**Il peut être intéressant de montrer l'architecture globale de votre application surtout si elle est complexe (micro service)**



Architecture micro service avec une gateway filter et un annuaire de service

## Recommandation de sécurité

Pour votre application, il est nécessaire de montrer vos mises en pratique liées aux recommandations de sécurité sur les 3 CCP.

Voici la liste non exhaustive des bonnes pratiques attendues par le jury :

**Validation des entrées** : Assurez-vous de toujours valider et filtrer toutes les entrées utilisateur, que ce soit les champs de formulaire, les paramètres d'URL.

**Protection contre les attaques de script (XSS)** : Évitez l'exécution de scripts non autorisés en appliquant l'échappement des caractères spéciaux lors de l'affichage des données dynamiques dans les pages HTML.(Cela peut être réalisé par des bibliothèques)

**Authentification et gestion des sessions** : Lorsque votre application nécessite une authentification, utilisez des mécanismes d'authentification robustes tels que OAuth, OpenID Connect ou JSON Web Tokens (JWT).

**Gestion des autorisations** : Mettez en place un système d'autorisation rigoureux pour contrôler l'accès aux fonctionnalités et aux données sensibles. Définissez des rôles et des permissions appropriés pour chaque utilisateur, en vous assurant que seuls les utilisateurs autorisés peuvent accéder aux ressources pertinentes.

**Sécurité des communications** : Utilisez des protocoles de communication sécurisés tels que HTTPS pour chiffrer les échanges de données entre le frontend et le backend. Cela empêche les attaquants d'intercepter ou de manipuler les données transitant sur le réseau.

**Gestion des erreurs** : Évitez de révéler des informations sensibles lors de l'affichage des messages d'erreur. Personnalisez les messages d'erreur pour fournir des informations génériques sans divulguer de détails sensibles sur l'application ou son infrastructure.

### 3. le déroulé de l'examen



L'épreuve est découpée en 3 phases :

<b>Présentation du projet</b>	<b>Entretien technique</b>	<b>Entretien final</b>
-------------------------------	----------------------------	------------------------

#### Présentation du projet

 **40 minutes** : c'est le temps imparti pour faire votre présentation.

Le délai de 35 minutes ne doit pas être pris à la légère. Si vous terminez vos simulations d'entraînement en moins de 35 minutes, cela peut indiquer un problème avec votre présentation, le fond ou la forme.


Au cours de ces 40 minutes, vous vous présenterez devant un jury composé de deux personnes (parfois trois). La présentation de votre powerpoint sera diffusée sur un projecteur fourni par le lieu de la session. Il est préférable d'utiliser votre ordinateur personnel pour avoir accès à tous les documents nécessaires et pour exécuter votre projet en local en cas de besoin, notamment si le projet n'a pas été déployé ou en cas d'anomalie.

En règle générale, le jury n'intervient pas durant ce temps. Il n'est pas question de perturber le candidat ou de le stresser. Vous serez alors dans un long monologue et il ne faudra pas non plus être décontenancé face au mutisme de ces derniers ou bien de leur occupation.

En général, pendant cette période, le jury ne doit pas intervenir. Cela peut perturber et stresser le candidat. Il est important de continuer à parler sans se laisser déconcentrer par le silence ou les actions des membres du jury.

A l'issue, vous entrerez dans la deuxième phase :

## Entretien technique

 **45 minutes** : c'est le temps imparti pour l'entretien technique

Le jury prendra le temps nécessaire pour discuter de votre projet avec vous et clarifier certains concepts. Cela peut inclure des questions sur les choix conceptuels que vous avez faits, des questions théoriques liées à votre formation et d'autres éléments pertinents.

L'objectif de l'entretien est de s'assurer que le candidat est bien l'auteur du projet, qu'il en comprend bien les tenants et les aboutissants, et qu'il a une maîtrise totale du code utilisé.

## Entretien final

 **20 minutes** : c'est le temps imparti pour l'entretien final

Il convient de souligner que l'entretien est très formel. L'objectif est de discuter de votre parcours professionnel et de vos objectifs futurs. Nous aimerions savoir si vous envisagez de continuer à vous former ou si vous avez déjà trouvé un poste.

## 4. Le powerpoint et l'oral

Véritable support visuel de votre présentation à travers lequel vous exposerez oralement vos arguments. Cependant, veuillez à éviter de copier-coller votre



mémoire sur votre présentation. Au lieu de cela, pensez à créer une infographie dynamique qui vous guidera tout au long de votre présentation, plutôt qu'un bloc de texte monotone que vous lirez simplement.

**Je vous suggère de réaliser votre présentation PowerPoint en deux parties.**

## 1er partie 15 minutes

- Introduction du projet en anglais

*(Astuce pour savoir quoi mettre dans l'intro : répondre aux questions du QQQQCP : Qui, Quoi, où, Quand, Comment, Pourquoi)*

- Conception fonctionnelle (avec diagramme de use case, au moins 1 diagramme de séquence, 1 diagramme de classe ou merise )
- Conception graphique (wireframe, charte graphique)
- Méthodologie et outils utilisés (agile, scrum, kanban, trello, etc..)
- Environnement technique (vs code, git, php, java, javascript...)



**La démonstration de l'application se fait idéalement à la fin de la partie 1 du powerpoint et dure 5 minutes.**

Durant la démonstration, il convient de créer un jeux de données afin de montrer la communication du Frontend avec le back-end ainsi que la persistance des données.

**Exemple :**

1. Créer un utilisateur
2. Connecter l'utilisateur
3. Afficher le site en version mobile pour montrer le côté responsif
4. Se rendre sur l'interface de la base de données (phpMyAdmin) pour montrer en temps réel l'insertion dans la base de données.
5. Effectuer une autre action en fonction du contexte de votre application (effectuer une réservation, ajouter un article au panier...)

## 2eme partie 20 minutes

Suite à la démonstration il est alors pertinent de montrer ces exemples de fonctionnalité épluchées avec des captures d'écran de code.

### **Il conviendra alors de montrer :**

1. la page du formulaire d'inscription côté front.
2. le code CSS qui gère le responsif (media queries ou classe bootstrap)
3. la partie back end qui gère la soumission du formulaire et qui effectue la requête à la bdd.

Répéter l'opération pour la connexion et la 3ème fonctionnalité montrée dans la démo.

- Faire une slide spécialement autour de la sécurité de l'application
- Montrer des exemples de recherches autonomes ou de veille technologique en anglais.
- Conclusion

Après votre prestation, le jury aura déjà une idée de votre niveau et de vos chances d'obtenir le titre. Par conséquent, plus votre présentation sera complète, moins vous aurez de questions à répondre devant le jury.

## 5. Les questions du jury

En premier lieu, le jury devra s'assurer que vous êtes bien l'auteur du projet. Il est en effet possible que le travail présenté ne soit pas le vôtre, mais plutôt celui d'un camarade, d'une session précédente ou même récupéré sur Github. Donc, leur première question sera de savoir **si vous êtes bien l'auteur du projet** et si vous **comprenez le contenu que vous avez écrit**.

Pour ce faire il pourra aborder plusieurs questions aussi bien théorique qu'en rapport avec votre projet.

Je vous propose ci dessous une liste de question déjà posée par thème :

### Questions théoriques POO

Qu'est ce qu'une classe ?

Quelle est la différence avec un Objet ?

Quels sont les 3 principes de la POO ?

Quels est l'intérêt d'utiliser le mot clé "static"

Expliquer l'héritage ?

La différences entre l'interface et les classes abstraites.

Expliquer le polymorphisme.

Comment sont détruit les objets ?

Pouvez vous me citer 2 design pattern ?

# Questions Conception

## **diagramme cas d'utilisation <use case> :**

Expliquer la notion d'extend et d'include.

Expliquer la notion d'acteur et leur position par rapport au système.

## **diagramme de séquence:**

Expliquer brièvement ce qu'est un diagramme de séquence.

Identifiez les acteurs, les objets, les messages et les lifelines (lignes de vie) représentés.

Expliquez un cas de scenario alternatif et sa représentation graphique.

## **diagramme de classe :**

Expliquer l'agrégation, la relation et la composition.

Expliquer les multiplicités et leur intérêts.

Expliquer le - et le + devant les propriétés/méthodes

## **Merise:**

Expliquer la différence entre le MCD et le MPD.

Expliquer les cardinalités et leur intérêts.

Expliquer les règles de passage du MCD vers le MPD.

Expliquer le sort des clé primaires dans le passage du MCD vers le MPD dans le cas @ManyToOne, @ManyToMany, @OneToOne

## **Multicouche :**

Expliquer ce qu'est une application multicouche.

De combien de couches est constituée votre application.

Quel est l'intérêt de découper l'application à travers différentes couches.

## Questions Persistance des données

Expliquer ce qu'est un ORM, quel est son intérêt ?

Expliquer le CRUD.

Lorsque vous travaillez avec une base de données relationnelle sans ORM, comment effectuez-vous les opérations CRUD ?

Pouvez vous me reproduire le schéma des différentes jointures SQL.

Ecrire une requête SQL de jointure entre 2 tables (ex clients et produits).

## Questions Sécurité

Citez 3 failles de sécurités.

Comment se protéger d'une injection SQL.

C'est quoi une attaque CSRF ?

A quoi sert un token JWT ?

Quelle est la différence entre hachage et cryptage ?